

REMARKS

Claims 1-16 are pending in the instant application. Of these, claims 7 and 12 have been rejected under 35 U.S.C. 102(b) as allegedly being anticipated by U.S. Patent No. 5,835,770, issued to Shum et al. (hereafter Shum '770). Claims 1-6 have been rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Shum '770, in view of the publication "MICROSOFT BACKOFFICE (TM) Administrator's Survival Guide," by Arthur Knowles, published by SAMS PUBLISHING (TM), 1996 (hereafter Knowles '96), and in further view of U.S. Patent No. 5,617,566 to Malcolm (hereafter Malcolm '566). Claims 8 and 13 have been rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Shum '770, in view of the U.S. Patent No. 4,166,290 issued to Furtman et al. (hereafter Furtman '290). Claims 9 and 14 have been rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Shum '770, in view of the U.S. Patent No. 5,892,898 issued to Fujii et al. (hereafter Fujii '898). Claims 10-11 and 15-16 have been rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Shum '770, in view of Knowles '96. Claims 1-16 have been canceled. New claims 17-33 have been added. The Applicant submits that claims 17-33 are in condition for allowance for at least the reasons provided herein.

The Applicant's new claims 17 and 24 respectively recite a method and storage medium for managing computer performance, comprising:

"for each task executing on a computer system, performing:  
capturing details relating to said execution while said task is active;  
and  
storing said details in an activity record; and  
upon detecting an error relating to a task:  
retrieving activity records associated with said task in which said  
error was detected; and

analyzing said activity records to determine a root cause of said error;

wherein said details in an activity record include:

a task name assigned by a user of said computer system;

and

a flag specifying whether said task was successfully completed."

None of the cited references (Shum '770, Knowles '96, Malcolm '566, Furman '290, and Fujii '898), hereinafter referred to collectively as 'references', recite those features. Specifically, none of the references recite capturing details for every task executing on a computer system while each task is active. Nor do the references recite storing the details in an activity record and retrieving activity records associated with a task for which an error has been detected. The references also do not recite assigning a task name to the task and specifying whether the task was successfully completed via a flag.

Shum '770 is directed to solving the problem of determining a current status of an application or program executing within a data processing system that would not otherwise be available to the user and presenting the status to a system user upon request. (col. 1, lines 25-33). Conversations (i.e., task progress information) occurring within the system are gathered and stored in the system (col. 3, lines 25-49). An inquiry instruction is activated by a user "to select the task progress tracing information that was stored, and this selected information, which may represent the state of the user's program, the state of communications with the user, identification of the communication partner, and the like is prepared for sending to the user..." (col. 3, lines 37-45).

Shum '770 further recites a monitoring function that gathers task progress tracing information that is recorded in one or more storage blocks (col. 5, lines 25-28). "The recorded information, preferably includes the type of database being accessed, communication line status, communication location identification, the stage of the user

ROUT20010026US1/TB2-0018

application being executed, etc. At any point after initiation of the user application, the user may, by employing a facility offered by this invention, address the storage blocks to determine the current status of any communication links, databases being used, or the status of the application program itself, to determine whether the application program is running, the identification of the services being accessed and the status of the communication lines" (col. 5, lines 28-39).

While Shum '770 recites some monitoring and storage capabilities, the similarities end there. Shum '770 does not teach or suggest an activity record including a task name for the task that is assigned by a user and a flag specifying whether the task was successfully completed. Moreover, Shum '770 does not teach or suggest creating an activity record for each execution of a task. Rather, Shum '770 simply stores the gathered information in one or more storage blocks. The activity records of the instant application provide information about each execution of a task. The task name assigned is used to retrieve all activity records that relate to each execution of the task. The flag further differentiates between successfully completed tasks and unsuccessfully completed tasks, which enable a distinction to be made among each of the activity records that are retrieved for the task. Thus, without a task name and without an activity record generated for 'each execution of a task', Shum '770 would not be capable of retrieving its stored information in the manner as recited in the instant application.

The Examiner suggests in the instant Action that Shum '770 recites a task name, namely, a status inquiry command that is entered by a user at a terminal (col. 6, lines 10-11). The Applicant respectfully disagrees and submits that the status inquiry command is a system command that triggers the execution of the inquiry facility. The inquiry facility, in turn, causes "the execution of the user's application program ...[to be] suspended temporarily...as the inquiry facility requests and receives *current information from an information storage bank* [such as] the status of the application program, connections being made, database information. (col. 6, lines 12-17) (emphasis added). The status inquiry command is clearly not equivalent to the task name as recited in the instant application. Moreover, Shum '770 does not recite analyzing the activity records to

POU920010026US1/1B2-0018

determine a root cause of an error.

Knowles '96 is an administrator survival guide for assisting system users in optimizing operating system performance. As indicated above, the Applicant submits that the limited operating system techniques exist for monitoring system performance, however, the administrator guide of the Knowles reference does not recite the features as provided in Applicant's claims 17 and 24. Accordingly, the Applicant submits that claims 17 and 24 are patentable over Knowles '96.

Malcolm '566 recites an auxiliary database management system. The Applicant has canceled claims directed to this feature and submits that the rejections based upon the auxiliary database are rendered moot.

Fujii '898 recites an event management system and storage media for identifying and logging event information. The event management system and storage media include an application for reporting an event message in response to an occurrence of a particular event (Abstract). While Fujii '898 recites identifying "the source of a service error and obtain[ing] detailed error information...[and] reporting [error] information as part of its error handling operation" (col. 1, lines 52-58), these activities are not associated with a task which, as recited in the claims 17 and 24, is further associated with activity records for each execution of the task. Accordingly, claims 17 and 24 are patentable over Fujii '898 for at least these reasons.

Applicant's claims 19 and 29 recite a contention analysis report generated utilizing details providing in activity records and environmental records. Furtman '290 recites performance monitoring equipment that connects monitoring hardware as if it were a peripheral device to a CPU channel by picking off signals on the channel, by monitoring combination of signals and sequences of signals and generating event codes which identify the combinations and sequences, by reducing the data picked off the channel in accordance with the programmable instructions for each peripheral device on the channel, and collecting packets of information dependence upon the event code generated" (col. 2, lines 15-25). While Furtman '290 also recites making "DASD measurements, such as: contention analysis." (col. 4, lines 30-31), this contention analysis

POU920010026US1/IB2-0018

is not performed in response to details provided in activity records and environmental records as recited in the instant application.

Accordingly, because none of the cited art references, individually or in combination, teach or suggest each element of the Applicant's claims 17 and 24, the Applicant respectfully submits that claims 17 and 24 are in condition for allowance. For at least these reasons, the Applicant requests reconsideration of the rejections. Claims 18-23 depend from claim 17 and claim 25 depends from claim 24. The Applicant further submits that dependent claims 18-23 and 25 are in condition for allowance for at least the reasons stated above.

Claims 26-28 recite a system for managing computer system performance. The Applicant submits that claims 26-28 are in condition for allowance for at least the reasons provided above with respect to claims 17 and 24. Claims 29-33 depend from what should now be allowable claims 26-28. Accordingly, for at least this reason, the Applicant submits that claims 29-33 are in condition for allowance.

Notwithstanding, Applicant's claim 18 and 25 respectively recite:

"for said each task executing on a computer system, performing:

capturing environmental information associated with computer system activity that occurs while said task is active;

creating an environmental record; and

storing said environmental information in said environmental record; and if said analyzing said activity records does not reveal a root cause of said error:

retrieving environmental records created between a last successful execution of a program associated with said task and a time in which said error occurred; and

analyzing said environmental records in conjunction with said activity records to determine a root cause of said error;

whereto said environmental information includes:

a name of said program; and

environmental flags operable for classifying said task."

None of the cited references recites these features. Specifically, none of the references teach or suggest capturing environmental information relating to computer system activity for each task executing on a computer system and creating an environmental record therefrom. Nor do the references recite retrieving environmental records created between a last successful execution of a program associated with said task and a time in which said error occurred. While, the Examiner correctly states that some environmental information may be passed from an operating system to an application, the Applicant submits that the instant application distinguishes between the activity conducted by an operating system and the activities recited in the instant application. The environmental information recited in the instant application is collected during execution of each task. An environmental record is generated for this task and stores the environmental information collected. The environmental record includes environmental flags for classifying a task and also includes a program name associated with the task. Thus, neither the references, nor existing operating systems generate environmental records for each task executing on a computer system, whereby the records include environmental flags for classifying a task and a program name associated with a task. Accordingly, for at least the reason that the cited art references do not recite each of the features of claims 18 and 25, the Applicant submits that claims 18 and 25 are in condition for allowance.

Applicant's claims 19 and 29 recite a contention analysis report generated utilizing details providing in activity records and environmental records. Furtman '290 recites performance monitoring equipment that connects monitoring hardware as if it were a peripheral device to a CPU channel by picking off signals on the channel, by monitoring combination of signals and sequences of signals and generating event codes which identify the combinations and sequences, by reducing the data picked off the channel in accordance with the programmable instructions for each peripheral device on the channel, and collecting packets of information dependence upon the event code generated" (col. 2, lines 15-25). While Furtman '290 also recites making "DASD measurements, such as: contention analysis..." (col. 4, lines 30-31), this contention

POU920010026US1/1B2-0018

analysis is not performed in response to details provided in activity records and environmental records as recited in the instant application. Accordingly, the Applicant submits that claims 19 and 29 are patentable over Furtman '290.

No new matter has been entered and no additional fees are believed to be required. However, if any fees are due with respect to this Amendment, please charge them to Deposit Account No. 09-0458 maintained by Applicant's Assignee.

Respectfully submitted,  
CANTOR COLBURN LLP

By Marisa J. Dubuc

Marisa J. Dubuc  
Registration No. 46,673  
Customer No. 23413

Date: April 27, 2004  
Address: 55 Griffin Road South, Bloomfield, CT 06002  
Telephone: (860) 286-2929